CPSC 1155 – Lab 1

Introductory Lab

Learning Objectives

At the end of this lab, you should:

- have installed either an IDE or a code editor and compiler
- be able to use a programming environment to write, compile, and run a C++ program
- submit a lab on Brightspace

Lab Reading / Viewing

1. Lab Guide video on Brightspace
2. Chapter 1 – Introduction to Programming

Lab Introduction

This lab provides you with an understanding of the programming tools you will use for your labs throughout the semester.

A C++ program (source code) is written as a simple text file, with no formatting, for which we need a text editor. The C++ source code then needs to be translated to an object code using a C++ compiler. The linker puts together the object code with other object codes (like libraries) into an executable that is then run. You will learn more about the details in Chapter 2.

You may either install and use an IDE or a code editor along with a compiler to write and execute your C++ programs. Online programming environments can be used to quickly test your programs, but you are not going to use them for completing the labs and assignments.

For C++ source filenames, it is preferred to use meaningful multiple-word names all in lowercase letters. Either separate the words with underscore or a capital letter. The filename ends in .cpp. For example, myFirstProgram.cpp or my_first_program.cpp.

Lab Instructions

A. Install an IDE or a code editor

You have the following options to write your programs. Make sure to view the Lab Guide video on Brightspace.

1. Code editor and compiler

   You may install a code editor such as Visual Studio Code, Sublime Text, Atom, and so on. Do your research, choose your preferred code editor, download, and install it on your device.

   If you are using a code editor, you also need to install a compiler. You may install MinGW-w64.

2. IDE

   You may install an IDE (Integrated Development Environment) such as Visual Studio Community. It is used for editing, compiling, linking, and running C++ programs all in one.

3. Online editors

   Online programming environments such as OnlineGDB, repl.ir, or C++ Shell can be used for testing your programs. However, they might have limited capabilities comparing to the above options.

B. Create a Simple C++ Program

In the following section, you will create a simple C++ program.

1. **Open a text editor application**: Open either your IDE or your code editor.
2. **Enter your first program**: Type in the following "printHello.cpp" program exactly as shown below. Pay close attention to the wording and indentation.

   Notice the comment header of the program with the required information. **Make it a habit to add a similar comment header to your programs.** Be sure to change the name to your name as author and write the correct date. This is important internal documentation.

   It is strongly recommended that you type in the code rather than simply copy and paste. This will help you learn the syntax and be able to fix your **syntax errors**.

```
//-------------------------------------------------------------
// Lab 1: Getting Started with C++
// printHello.cpp: The program prints Hello World! then Hello Again.
//
// Author: Your Name                         Date: Current Date
//-------------------------------------------------------------
#include <iostream>
using namespace std;

int main() {
   // This program outputs 2 messages
   cout << "Hello World!" << endl;
   cout << "Hello Again." << endl;

   return 0;                     // Return to OS and indicate no errors
} // End main
```

3. **Save your program**: Create a folder on your computer called CPSC1155. Create a subfolder for Lab1. Save this simple program as "printHello.cpp" in this folder.

   When entering a long program, save your file often so that you do not lose what you have typed in. Always save your file before leaving a computer.

   You may save copies of your files on Langara myfiles (https://myfiles.langara.bc.ca/WebInterface/login.html) to make sure you never lose your files!
4. **Compile your program**: Once you have saved your program, compile it by invoking the C++ language compiler. Look in the menus and find an option to compile and run the program (Make sure to watch the Lab Guide video on Brightspace to learn how to run your C++ programs).

If there are any syntax errors, they will be reported in a window below the source code. Select the error with the mouse and it should take you to the general area in the editor where the error is.

You will need to determine what errors you might have made and fix them in the source code displayed in the main editor window. You will need to save your file again after any changes and recompile to make sure your changes are scanned by the compiler as it reads your C++ source code file. If you still have errors, you will need to repeat this sequence until you no longer have any syntax or other compiler detected errors.

When compilation is successful, the compiler will compile and link the libraries and create a new file called "printHello.exe" in the same working directory. It is a Windows executable file.

5. **Run your program**: Once you have compiled your program into C++ executable code, contained within the ".exe" file, you can run your program within the editor. Your program executes and its output appears in the small window below the edit window. Alternatively, you can see the output on a black screen by running the ".exe" file.

6. **Take a screenshot**: Press *PrtSc* (or *PrintScreen*) button on your keyboard to take a screenshot of the screen. The key is often located on the top row of the keyboard, to the right of the function keys. Sometimes, the label indicates that you must simultaneously press the *Shift* or *Fn* key. It appears that nothing happens, but the computer takes a picture of the entire screen and saves it in the system clipboard.

   Alternatively, you may use *Alt+PrtSc* (hold *Alt* while pressing *PrtSc*) to take a screenshot of the active window.

   Open the **Paint** program on your computer. Paste (press *Ctrl+v*) your screenshot in Paint. Save your file with the following name: "printHello.jpg" under the Lab1 folder.

7. **Test your program**: Write your answer to the following question in "printHello.cpp" as a comment:

   a. Comment out the line `#include <iostream>` (add // in front of the line). What happens? What is the error description?

C. Create a Simple C++ Program with onlineGDB

In the following section, you will write and compile a C++ program using **onlineGDB**.

1. **Open an online text editor**: Open onlineGDB.com in your browser. Select C++ from the list of languages.

2. **Enter your program**: Type the following program "printName.cpp" in the provided area.

```
//-------------------------------------------------------------
// Lab 1: Working with an online tool
// printName.cpp: This program asks for your name and greets you!
//
// Author: Your Name                         Date: Current Date
//-------------------------------------------------------------
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
   // Declare a variable for your name
   string name;

   // Ask for the input
   cout << "Please enter your name: ";
   cin >> name;

   // A greeting is printed
   cout << "Hello " << name << endl;

   return 0;
} // End main
```

3. **Save your program**: In order to save your program, you need an online account with Google+, Facebook, or GitHub. If you do not have those accounts, you do not need to save your file now.

   Note: You can always cut and paste your code in a **Notepad** file or any code editor and save it as a .cpp file on a local drive. Save this file as "printName.cpp".

4. **Run your program**: Click "Run" and you will see the results in the lower pane.

5. **Take a screenshot**: Press *Alt+PrtSc* (or *PrintScreen*) button on your keyboard to take a screenshot of the screen and paste (press *Ctrl+v*) it in Paint program. Save your file with the following name: "printName.jpg" under the Lab1 folder.

Lab Submission

Create a zip file containing two .cpp files and two .jpg files that you created in this lab. Name your zip file as "yourName_Lab1.zip" and submit on Brightspace by the due time.